

# Python: module vcs.projection

## vcs.projection

[index](#)

# Projection (Proj) module

### Modules

[vcs.VCS\\_validation\\_functions](#) [vcs.vcs](#)

### Classes

builtin .object  
Proj

class **Proj**(builtin .object)

Class: Proj # Projection

Description of Proj Class:

The projection secondary method (Proj) is used when plotting 2D or 3D data. It provides a way to project from lon/lat coord to another mapping system (lambda, phi).

This class is used to define a projection table entry used in VCS. It can be used to change some or all of the attributes in an existing projection table entry.

Description of parameters (from USGS Documentation)

Projection Transformation Package Projection Parameters

Code & Projection Id	Array Element					
	1	2	3	4	5	6
0 Geographic						
1 U T M	Lon/Z	Lat/Z				
2 State Plane						
3 Albers Equal Area	SMajor	SMinor	STDPR1	STDPR2	CentMer	Origin
4 Lambert Conformal C	SMajor	SMinor	STDPR1	STDPR2	CentMer	Origin
5 Mercator	SMajor	SMinor			CentMer	TrueS
6 Polar Stereographic	SMajor	SMinor			LongPol	TrueS
7 Polyconic	SMajor	SMinor			CentMer	Origin
8 Equid. Conic A	SMajor	SMinor	STDPAR		CentMer	Origin

	Equid. Conic B	SMajor   SMinor   STDPR1   STDPR2   CentMer   Orig
9	Transverse Mercator	SMajor   SMinor   Factor     CentMer   Orig
10	Stereographic	Sphere         CentLon   Cente
11	Lambert Azimuthal	Sphere         CentLon   Cente
12	Azimuthal	Sphere         CentLon   Cente
13	Gnomonic	Sphere         CentLon   Cente
14	Orthographic	Sphere         CentLon   Cente
15	Gen. Vert. Near Per	Sphere     Height     CentLon   Cente
16	Sinusoidal	Sphere         CentMer
17	Equirectangular	Sphere         CentMer   TrueS
18	Miller Cylindrical	Sphere         CentMer
19	Van der Grinten	Sphere         CentMer   Orig
20	Hotin Oblique Merc A	SMajor   SMinor   Factor     Orig
	Hotin Oblique Merc B	SMajor   SMinor   Factor   AziAng   AzmthPt   Orig
21	Robinson	Sphere         CentMer
22	Space Oblique Merc A	SMajor   SMinor     IncAng   AscLong
	Space Oblique Merc B	SMajor   SMinor   Satnum   Path
23	Alaska Conformal	SMajor   SMinor
24	Interrupted Goode	Sphere
25	Mollweide	Sphere           CentMer
26	Interrupt Mollweide	Sphere
27	Hammer	Sphere           CentMer
28	Wagner IV	Sphere           CentMer
29	Wagner VII	Sphere           CentMer
30	Oblated Equal Area	Sphere     Shapem   Shapen   CentLon   Cente

Projection Transformation Package Projection Parameters elements  
continued

Code & Projection Id	Array Element					
	9	10	11	12	13	
0 Geographic						
1 U T M						
2 State Plane						
3 Albers Equal Area						
4 Lambert Conformal C						
5 Mercator						
6 Polar Stereographic						
7 Polyconic						
8 Equid. Conic A	zero					
	Equid. Conic B	one				
9 Transverse Mercator						
10 Stereographic						
11 Lambert Azimuthal						
12 Azimuthal						
13 Gnomonic						
14 Orthographic						
15 Gen. Vert. Near Per						

16	Sinusoidal							
17	Equirectangular							
18	Miller Cylindrical							
19	Van der Grinten							
20	Hotin Oblique Merc A	Long1	Lat1	Long2	Lat2	zero		
	Hotin Oblique Merc B					one		
21	Robinson							
22	Space Oblique Merc A	PSRev	LRat	PFlag		zero		
	Space Oblique Merc B					one		
23	Alaska Conformal							
24	Interrupted Goode							
25	Mollweide							
26	Interrupt Mollweide							
27	Hammer							
28	Wagner IV							
29	Wagner VII							
30	Oblated Equal Area	Angle						

---

where

Lon/Z	Longitude of any point in the UTM zone or zero. If zero, a zone code must be specified.
Lat/Z	Latitude of any point in the UTM zone or zero. If zero, a zone code must be specified.
SMajor	Semi-major axis of ellipsoid. If zero, Clarke 1866 is assumed.
SMinor	Eccentricity squared of the ellipsoid if less than zero, if zero, a spherical form is assumed, or if greater than zero, the semi-minor axis of ellipsoid.
Sphere	Radius of reference sphere. If zero, 6370997 meters
STDPAR	Latitude of the standard parallel
STDPR1	Latitude of the first standard parallel
STDPR2	Latitude of the second standard parallel
CentMer	Longitude of the central meridian
OriginLat	Latitude of the projection origin
FE	False easting in the same units as the semi-major axis
FN	False northing in the same units as the semi-major axis
TrueScale	Latitude of true scale
LongPol	Longitude down below pole of map
Factor	Scale factor at central meridian (Transverse Mercator) or center of projection (Hotine Oblique Mercator)
CentLon	Longitude of center of projection
CenterLat	Latitude of center of projection
Height	Height of perspective point
Long1	Longitude of first point on center line (Hotine Oblique Mercator, format A)
Long2	Longitude of second point on center line (Hotine Oblique Mercator, format A)
Lat1	Latitude of first point on center line (Hotine Oblique Mercator, format A)
Lat2	Latitude of second point on center line (Hotine Oblique Mercator, format A)

AziAng	Azimuth angle east of north of center line (Hotine Oblique Mercator, format B)
AzmthPt	Longitude of point on central meridian where azimuth (Hotine Oblique Mercator, format B)
IncAng	Inclination of orbit at ascending node, counter-clockwise from equator (SOM, format A)
AsCLong	Longitude of ascending orbit at equator (SOM, format A)
PSRev	Period of satellite revolution in minutes (SOM, format A)
LRat	Landsat ratio to compensate for confusion at northern end of orbit (SOM, format A -- use 0.5201613)
PFlag	End of path flag for Landsat: 0 = start of path, 1 = end of path (SOM, format A)
Satnum	Landsat Satellite Number (SOM, format B)
Path	Landsat Path Number (Use WRS-1 for Landsat 1, 2 and 3, WRS-2 for Landsat 4, 5 and 6.) (SOM, format B)
Shapem	Oblated Equal Area oval shape parameter m
Shapen	Oblated Equal Area oval shape parameter n
Angle	Oblated Equal Area oval rotation angle

#### NOTES

Array elements 14 and 15 are set to zero

All array elements with blank fields are set to zero

All angles (latitudes, longitudes, azimuths, etc.) are entered in degrees/ minutes/ seconds (DDDDMMSSS.SS) format

The following notes apply to the Space Oblique Mercator A project:

A portion of Landsat rows 1 and 2 may also be seen as parts of rows 246 or 247. To place these locations at rows 246 or 247, set the path flag (parameter 11) to 1--end of path. This flag defaults to 0.

When Landsat-1,2,3 orbits are being used, use the following values for the specified parameters:

Parameter 4	099005031.2
Parameter 5	128.87 degrees - (360/251 * path number) in packed DMS format
Parameter 9	103.2669323
Parameter 10	0.5201613

When Landsat-4,5 orbits are being used, use the following values for the specified parameters:

Parameter 4	098012000.0
Parameter 5	129.30 degrees - (360/233 * path number) in packed DMS format
Parameter 9	98.884119
Parameter 10	0.5201613

Note: In vcs angles can be entered either in DDDMMSSS or regular angles.

```

Other Useful Functions:
    a=vcs.init()                      # Constructor
    a.show('projection')               # Show predefined project

Example of Use:
    a=vcs.init()

    To Create a new instance of projection use:
        p=a.createprojection('new','quick') # Copies content of 'quick'
        p=a.createprojection('new')         # Copies content of 'default'

    To Modify an existing projection use:
        p=a.getprojection('lambert')

        p.list()                         # Will list all the projection attributes
        p.type='lambert'
        p.parameters=[1.e20,1.e20,1.e20,1.e20,1.e20,1.e20,1.e20,1.e20,1.e20,1.e20]
        iso=x.createisoline('new')
        iso.projection=p
        #or
        iso.projection='lambert'

```

Methods defined here:

```

__init__(self, parent, Proj_name=None, Proj_name_src='default', createProj=0)
    ######
    #
    # Initialize the projection attributes.
    #
    #####
checkPP(self, name, value)

list(self)
    #####
    #
    # List out projection secondary method members (attributes).
    #
    #####
rename = renameProj(self, old_name, new_name)
    #####
    #
    # Function:      renameProj
    #
    # Description of Function:
    #           Private function that renames the name of an existing
    #           secondary method.
    #
    #
    # Example of Use:
    #     renameProj(old_name, new_name)
    #           where: old_name is the current name of project

```

```

#                                     new_name is the new name for the project
#
#####
#script(self, script_filename=None, mode=None)
Function:      script                                         # Calls _vcs.s
Description of Function:
Saves out a projection secondary method in Python form to a
designated file.

Example of Use:
script(scriptfile_name, mode)
where: scriptfile_name is the output name of the script file.
mode is either "w" for replace or "a" for append.

a=vcs.init()
p=a.createprojection('temp')
p.script('filename.py')                                # Append to a file "filename.py"
p.script('filename','w')

```

---

Properties defined here:

***angle***

```

get">get = _getangle(self)
set">set = _setangle(self, value)

```

***azimuthalangle***

```

get">get = _getazimuthalangle(self)
set">set = _setazimuthalangle(self, value)

```

***azimuthallongitude***

```

get">get = _getazimuthallongitude(self)
set">set = _setazimuthallongitude(self, value)

```

***centerlatitude***

```

get">get = _getcenterlatitude(self)
set">set = _setcenterlatitude(self, value)

```

***centerlongitude***

```

get">get = _getcenterlongitude(self)
set">set = _setcenterlongitude(self, value)

```

***centralmeridian***

```

get">get = _getcentralmeridian(self)
set">set = _setcentralmeridian(self, value)

```

***factor***

```

get">get = _getfactor(self)
set">set = _setfactor(self, value)

```

*falseeast*  
*get*">*get* = \_getfalseeast(self)  
*set*">*set* = \_setfalseeast(self, value)

*falsenorthing*  
*get*">*get* = \_getfalsenorthing(self)  
*set*">*set* = \_setfalsenorthing(self, value)

*height*  
*get*">*get* = \_getheight(self)  
*set*">*set* = \_setheight(self, value)

*landsatcompensationratio*  
*get*">*get* = \_getlandsatcompensationratio(self)  
*set*">*set* = \_setlandsatcompensationratio(self, value)

*latitude1*  
*get*">*get* = \_getlatitude1(self)  
*set*">*set* = \_setlatitude1(self, value)

*latitude2*  
*get*">*get* = \_getlatitude2(self)  
*set*">*set* = \_setlatitude2(self, value)

*longitude1*  
*get*">*get* = \_getlongitude1(self)  
*set*">*set* = \_setlongitude1(self, value)

*longitude2*  
*get*">*get* = \_getlongitude2(self)  
*set*">*set* = \_setlongitude2(self, value)

*name*  
*get*">*get* = \_getname(self)  
*set*">*set* = \_setname(self, value)

*orbitinclination*  
*get*">*get* = \_getorbitinclination(self)  
*set*">*set* = \_setorbitinclination(self, value)

*orbitlongitude*  
*get*">*get* = \_getorbitlongitude(self)  
*set*">*set* = \_setorbitlongitude(self, value)

*originlatitude*  
*get*">*get* = \_getoriginlatitude(self)  
*set*">*set* = \_setoriginlatitude(self, value)

*parameters*  
*get*">*get* = \_getparameters(self)  
*set*">*set* = \_setparameters(self, value)

***path***  
*get">get = \_getpath(self)*  
*set">set = \_setpath(self, value)*

***pathflag***  
*get">get = \_getpathflag(self)*  
*set">set = \_setpathflag(self, value)*

***satellite***  
*get">get = \_getsatellite(self)*  
*set">set = \_setsatellite(self, value)*

***satelliterevolutionperiod***  
*get">get = \_getsatelliterevolutionperiod(self)*  
*set">set = \_setsatelliterevolutionperiod(self, value)*

***shapem***  
*get">get = \_getshapem(self)*  
*set">set = \_setshapem(self, value)*

***shapen***  
*get">get = \_getshapen(self)*  
*set">set = \_setshapen(self, value)*

***smajor***  
*get">get = \_getsmajor(self)*  
*set">set = \_setsmajor(self, value)*

***sminor***  
*get">get = \_getsminor(self)*  
*set">set = \_setsminor(self, value)*

***sphere***  
*get">get = \_getsphere(self)*  
*set">set = \_setsphere(self, value)*

***standardparallel***  
*get">get = \_getstandardparallel(self)*  
*set">set = \_setstandardparallel(self, value)*

***standardparallel1***  
*get">get = \_getstandardparallel1(self)*  
*set">set = \_setstandardparallel1(self, value)*

***standardparallel2***  
*get">get = \_getstandardparallel2(self)*  
*set">set = \_setstandardparallel2(self, value)*

***subtype***  
*get">get = \_getsubtype(self)*  
*set">set = \_setsubtype(self, value)*

***truescale***

```
get">get = _gettruescale(self)
set">set = _settruescale(self, value)
```

***type***

```
get">get = _gettype(self)
set">set = _settype(self, value)
```

Data and other attributes defined here:

***\_\_slots\_\_*** = ['s\_name', 'setmember', 'smajor', 'sminor', 'centralmeridian', 'truescale', 'falseeastings', 'falseeastings', 'azimuthlongitude', 'longitude1', 'longitude2', 'latitude1', 'latitude2', 'subtype', 'orbitinclination', 'orbitinclination']

***azimuthlongitude*** = <member 'azimuthlongitude' of 'Proj' objects>

***parent*** = <member 'parent' of 'Proj' objects>

***s\_name*** = <member 's\_name' of 'Proj' objects>

***setmember*** = <member 'setmember' of 'Proj' objects>

## Functions

***getProjmember*(self, member)**

***renameProj*(self, old\_name, new\_name)**

```
#####
#
# Function:      renameProj
#
# Description of Function:
#     Private function that renames the name of an existing proj
#     secondary method.
#
#
# Example of Use:
#     renameProj(old_name, new_name)
#             where: old_name is the current name of projection
#                     new_name is the new name for the projection
#
#####
#
```

***setProjmember*(self, member, value)**

```
#####
#
# Function:      setProjmember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can-
```

```

#           set to 0, then this function does nothing.
#
#
# Example of Use:
#     setProjmember(self,name,value)
#           where: self is the class (e.g., Proj)
#                   name is the name of the member that is being
#                   value is the new value of the member (or att
#
#####
setmember = setProjmember(self, member, value)
#####
#
# Function:      setProjmember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:
#     setProjmember(self,name,value)
#           where: self is the class (e.g., Proj)
#                   name is the name of the member that is being
#                   value is the new value of the member (or att
#
#####

```

## Data

**StringTypes** = (<type 'str'>, <type 'unicode'>)